

# Wireless Deployer

## Guía de referencia de paquetes

Introducción .....	3
Package Descriptor .....	3
Sección PACKAGE:.....	3
Type .....	3
Id .....	4
Version.....	4
Name.....	4
FileList .....	4
UninstallList.....	4
Persistent.....	5
Sección INFO: .....	5
Description.....	5
Platform.....	5
Model .....	6
Vendor.....	6
Sección APPLICATION:.....	6
StartExe.....	6
Sección CONFIG:.....	7
Filelist .....	7
Ejemplo de archivo descriptor de un paquete de aplicación (application package descriptor): .....	7
Ejemplo de un archivo descriptor de un paquete de configuración (configuration package descriptor): .....	8
FileList .....	9
Sección DOWNLOAD: .....	9
Sección MAIN: .....	9
Secciones ACTION: .....	9
Variables de la Unidad.....	11
WDP Files.....	12
Sección MAIN: .....	13
Sección ACTION:.....	13
Guía de acciones .....	14
ACTION-ZIP .....	15
ACTION-CPY .....	16
ACTION-ATTRIB.....	16
ACTION-REG .....	17
ACTION-DLL .....	17
ACTION-EXE .....	17
ACTION-WAIT-EXE.....	18
ACTION-REGEXP.....	19
ACTION-DEL .....	20
ACTION-REMOVE .....	20

ACTION-KILL.....	21
ACTION-DELAY.....	21
ACTION-BOOT .....	22
Apéndice A: creación de archivos CPY.....	23

## Introducción

Este documento describe los archivos involucrados en la descarga de paquetes con WireLess Deployer:

- el **archivo descriptor del paquete** (Id-Version.pckg)
- el **archivo de instalación** (el nombre de este archivo es variable)
- el **archivo de desinstalación** (el nombre de este archivo es variable)

Estos archivos se pueden modificar con el Packet Express, pero también pueden modificarse a mano (no recomendado). La versión actual de Packet Express sólo permite construir **paquetes de aplicación**.

## Package Descriptor

Este archivo –ubicado en “X:\Program Files\WireLessDeployer\Packages”– contiene información acerca del paquete.

El nombre de este archivo está compuesto por el número de Id y de versión del paquete: *Id-Version.pckg*

La *Id* y la *Versión* **deben coincidir** con la Id y la Versión de la sección PACKAGE, de lo contrario, el paquete no será válido.

Hay dos tipos de paquetes: paquetes de aplicación y paquetes de configuración.

- Los archivos que describen paquetes de **aplicación** contienen las secciones PACKAGE, INFO y APPLICATION.
- Los archivos que describen paquetes de **configuración** contienen las secciones PACKAGE, INFO y CONFIG.

### Sección PACKAGE:

#### Type

Type = A | C

A = aplicación

C = configuración

Ejemplo:

Type = A

### **Id**

Identificación.

La combinación Id-Versión identifica un paquete de manera unívoca. Esta Id también es el nombre de la carpeta y del archivo del paquete. Es obligatorio que esta Id contenga sólo caracteres permitidos para carpetas y archivos de Windows.

### **Version**

Version = Major-minor-revision.

Use sólo dígitos, separados por guiones.

Ejemplo:

Version = 1-2-3

### **Name**

Nombre del paquete. Si se trata de un paquete de aplicación, el nombre del paquete es mostrado al usuario en la Consola WDp y en el Launcher.

### **FileList**

Nombre del archivo de instalación. Este archivo contiene una lista de archivos a descargar y de acciones a ejecutar (ver **FileList**). Este archivo es solicitado al instalar el paquete.

Ejemplo:

FileList = install.lstdnl

### **UninstallList**

Nombre del archivo de desinstalación. Este archivo puede contener una lista de archivos a descargar y de acciones a ejecutar (ver **FileList**). La UninstallList es solicitada al desinstalar el paquete.

Ejemplo:

FileList = uninstall.lstdnl

## Persistent

persistent = yes | no

Permite especificar si la información del paquete será persistente al coldboot. Para que los archivos de un paquete sean persistentes, el contenido del paquete **debe** ser almacenado en una **carpeta flash persistente** al cold-boot. Use ACCIONES, como [ACTION-CPY] para copiarlo en una carpeta flash (ver la **Guía de Acciones**) o directamente instale los archivos en la carpeta flash.

## Sección INFO:

Esta sección es únicamente informativa ya que es utilizada para mostrar información acerca del paquete en la Consola WDP. Ninguno de estos campos será utilizado en la PDA.

## Description

Descripción de paquete. Esta opción sólo tiene fines informativos.

## Platform

WINCE410 | WINCE420 | WINCE50 | WINCE2003 | WINCE2005

Plataforma en la que será instalado el paquete.

WINCE410: Windows CE 4.10

WINCE420: Windows CE 4.20

WINCE50: Windows CE 5.0

WINCE2003: Windows Mobile 2003 / Pocket PC 2003

WINCE2005: Windows Mobile 5.0

La Consola WDP y el Agente usan el valor de la plataforma en las colecciones y las fórmulas para evaluar condiciones de descarga de paquetes.

Es recomendado, aunque no obligatorio, que esta información se corresponda con la versión de sistema operativo que efectivamente tienen las PDAs que descargarán el paquete. El Cliente no verifica este valor.

Ejemplo:

Platform = WINCE50

### Model

Nombre del modelo o nombre estándar de la PDA.

La Consola WDP y el Agente usan este valor en las colecciones y las fórmulas para evaluar condiciones de descarga de paquetes.

Es recomendado, aunque no obligatorio, que esta información se corresponda con el modelo que efectivamente tienen las PDAs que descargarán el paquete. El Cliente no verifica este valor.

Ejemplo:

Model = mc9090

### Vendor

Especifica quién hizo el paquete. Esta opción sólo tiene fines informativos.

Ejemplo:

vendor = Softogo SA

### Sección APPLICATION:

Esta sección es opcional y es utilizada sólo en paquetes de aplicación ejecutados por el Launcher.

### StartExe

Camino completo a un ícono o a un archivo ejecutable en la PDA.

Vea la sección [ACTION-EXE] en el párrafo **WDP files** donde se explica el funcionamiento de esta acción.

El ícono mostrado en el Launcher es extraído de este archivo ejecutable. Si no hay ningún ícono disponible o si el archivo no es un exe, se mostrará un ícono estándar.

Ejemplo:

StartExe = \windows\calc.exe

## Sección CONFIG:

Esta sección es opcional y es utilizada sólo en paquetes de configuración. Permite al usuario modificar archivos de paquetes en la Consola WDp. Los archivos contenidos en esta lista no son descargados en la PDA.

### Filelist

Nombre del archivo que contiene la lista de configuración.

Ejemplo:

Filelist = config.lstcfg

### Ejemplo de archivo descriptor de un paquete de aplicación (application package descriptor):

```
# APP_SETUP-1-0-0.pkg

[PACKAGE]
type=A
id= APP_SETUP
name= APP setup
version=1-0-0
filelist=install.lstdn1
uninstalllist=uninstall.lstdn1
persistent=yes

[INFO]
description= APP application setup
platform=WINCE2005
model=MC9090
vendor=Softogo Packet Express

[APPLICATION]
StartExe= \windows\app.exe
```

**Ejemplo de un archivo descriptor de un paquete de configuración  
(configuration package descriptor):**

```
# APP_CONFIG-1-0-0.pckg

[PACKAGE]
type=c
id= APP_CONFIG
name= APP configuration
version=1-0-0
filelist=install.lstdnl
uninstalllist=uninstall.lstdnl
persistent=yes

[INFO]
description= APP configuration
platform=WINCE2005
model=MC9090
vendor=Softogo Packet Express

[CONFIG]
filelist = config.lstcfg
```

## FileList

Este archivo contiene la lista de archivos a descargar y un conjunto de acciones a ejecutar luego del proceso de descarga.

El archivo está compuesto por tres secciones: MAIN, DOWNLOAD y ACTIONS.

### Sección MAIN:

```
[main]
sequentialActions = yes | no
```

La opción *SequentialActions* permite ejecutar las acciones en el orden en que fueron escritas. Si esta opción no es activada o no es especificada, las acciones se ejecutarán siguiendo un orden predeterminado (ver **Secciones ACTION**).

### Sección DOWNLOAD:

Define los archivos que serán descargados.

```
[download]
filecount = número de líneas (NN+1)
file00    = archivo fuente > archivo de destino
...
fileNN    = fuente > destino
```

La fuente y el destino deben estar separados por un signo ">".

Fuente: camino del archivo en el servidor.

Destino: camino completo del archivo en la PDA.

No es necesario usar comillas para la fuente ni el destino.

### Secciones ACTION:

Estas secciones permiten definir un conjunto de acciones a ejecutar luego del proceso de descarga.

Las acciones soportadas son:

ACTION-ZIP, ACTION-CPY, ACTION-ATTRIB, ACTION-REG, ACTION-DLL, ACTION-EXE, ACTION-REGEXP, ACTION-DEL, ACTION-REMOVE, ACTION-KILL, ACTION-DELAY and ACTION-BOOT.

Vea la **Guía de acciones** para una descripción detallada de cada acción.

Existen dos modos de procesamiento de acciones: secuencial y no secuencial.

En el modo no secuencial, las acciones son procesadas en el siguiente orden:

- 1) ACTION-ZIP
- 2) ACTION-CPY
- 3) ACTION-ATTRIB
- 4) ACTION-REG
- 5) ACTION-DLL
- 6) ACTION-EXE
- 7) ACTION-REGEXP
- 8) ACTION-DEL
- 9) ACTION-REMOVE
- 10) ACTION-KILL
- 11) ACTION-DELAY
- 12) ACTION-BOOT

Para utilizar el modo no secuencial, desactive la opción *sequentialActions* en la sección MAIN.

En el modo secuencial, las acciones son procesadas en el orden en que fueron escritas, a excepción de ACTION-BOOT, la cual siempre se ejecuta al final, sin importar su ubicación en el archivo. Para usar este modo de ejecución, active la opción *sequentialActions* en la sección MAIN.

## Variables de la Unidad

Existe un conjunto de variables que pueden ser utilizadas en la sección DOWNLOAD de un archivo FileList. Estas variables caracterizan una unidad.

Las variables son:

```
%HOSTNAME%
%IP%
%MAC%
%MUID%
%UID%
%MODEL%
%PLATFORM%
```

El cliente busca estas variables en la sección DOWNLOAD y las reemplaza por el valor real. La búsqueda es realizada en la parte "fuente" de la sección, es decir, en el camino del archivo remoto.

Por ejemplo, si una unidad tiene estos valores:

```
HOSTNAME = PDA001
IP       = 192.168.1.9
MAC      = AA:BB:CC:DD:EE:FF
MUID     = ABC123DEF2345
UID      = john
MODEL   = MC9090
PLATFORM = WINCE50
```

y un archivo FileList de un paquete contiene estos archivos:

```
[DOWNLOAD]
filecount = 4
file00    = PDA\App\%MODEL%\app.exe > \Application\App\app.exe
file01    = PDA\Cfg\%HOSTNAME%.cfg > \Application\App\app.cfg
file02    = PDA\%UID%\file1.txt > \Application\App\file1.txt
file03    = PDA\%UID%\file2.txt > \Application\App\file2.txt
```

el cliente WDP reemplazará las variables de la siguiente manera:

```
PDA\App\%MODEL%\app.exe      PDA\App\MC9090\app.exe
PDA\Cfg\%HOSTNAME%.cfg      PDA\Cfg\PDA001.cfg
PDA\%UID%\file1.txt         PDA\john\file1.txt
PDA\%UID%\file2.txt         PDA\john\file2.txt
```

## WDP Files

Los archivos .wdp contienen una lista de acciones a ejecutar cuando se inicia la PDA. El cliente busca estos archivos en la carpeta flash de la PDA al arrancar. Una vez que los archivos han sido procesados, son ejecutados uno por uno. Existen varios niveles de archivos wdp:

.wdp, .wdp0, .wdp1, .wdp2, .wdp3, .wdp4 y .wdp5

Luego de un **coldboot**, se procesan los archivos .wdp, .wdp0, .wdp1, .wdp2, .wdp3, .wdp4 y .wdp5.

Luego de un **warmboot**, se procesan los archivos .wdp1, .wdp2, .wdp3, .wdp4 y .wdp5.

Los archivos son ejecutados en orden numérico ascendente, es decir, el primero en ejecutarse será wdp y el último será wdp5. Si dos o más archivos tienen el mismo número (por ejemplo "cfg.wdp2" y "app.wdp2") el orden es indistinto.

Pueden existir varios ACTION-BOOT en el procesamiento de archivos wdp. Sin embargo, sólo se ejecutará uno de ellos. La elección del ACTION-BOOT que se ejecute depende de los parámetros *type* e *immediate* y sigue las siguientes reglas:

- Si *immediate=yes*, la acción es ejecutada inmediatamente después del procesamiento del archivo wdp.
- Si se encuentran tanto COLDBOOT como WARMBOOT, se ejecuta sólo COLDBOOT.
- Si se encuentran varios WARMBOOT, se ejecuta el que tenga mayor delay.
- Si se encuentran varios COLDBOOT, se ejecuta el que tenga mayor delay.

La acción ACTION-BOOT debe usarse con precaución, ya que puede hacer que la PDA entre en un loop infinito. Por lo tanto, se recomienda no usar esta acción en archivos wdp que se ejecuten al realizar un warmboot (por ejemplo, wdp2, wdp3, wdp4 y wdp5).

El cliente Wdp puede ser usado como **instalador** ejecutando un archivo wdp5. Si el cliente encuentra un archivo wdp5 se cerrará aunque el archivo esté vacío o falle una acción.

El cliente Wdp crea un archivo de log ("\\WdpInstallerLog.txt") donde se escriben las acciones. Este archivo puede ser usado para el debogeo. Los archivos wdp constan de dos secciones: MAIN y ACTIONS.

## Sección MAIN:

[main]  
sequentialActions = yes | no  
autodelete = yes | no  
delayBeforeRunning = tiempo de demora en milisegundos

La opción *SequentialActions* permite ejecutar las acciones en el orden en que fueron escritas. Si esta opción no es activada o no es especificada, las acciones se ejecutarán siguiendo un orden predeterminado (ver Secciones ACTION).

La opción *Autodelete* permite borrar el archivo wdp una vez que se han ejecutado todas las acciones.

La opción *DelayBeforeRunning* permite definir la duración de la pausa que se efectuará antes de la ejecución de las acciones.

## Sección ACTION:

Define un conjunto de acciones a ejecutar.  
Las acciones soportadas son:

ACTION-ZIP, ACTION-CPY, ACTION-ATTRIB, ACTION-REG, ACTION-DLL, ACTION-EXE, ACTION-WAIT-EXE, ACTION-REGEXP, ACTION-DEL, ACTION-REMOVE, ACTION-KILL, ACTION-DELAY and ACTION-BOOT.

Vea la **Guía de acciones** para una descripción detallada de cada acción.

Existen dos modos de procesamiento de acciones: secuencial y no secuencial.

En el modo no secuencial, las acciones son procesadas en el siguiente orden:

- 13) ACTION-ZIP
- 14) ACTION-CPY
- 15) ACTION-ATTRIB
- 16) ACTION-REG
- 17) ACTION-DLL
- 18) ACTION-EXE
- 19) ACTION-REGEXP
- 20) ACTION-DEL
- 21) ACTION-REMOVE
- 22) ACTION-KILL
- 23) ACTION-DELAY

## 24) ACTION-BOOT

Para utilizar el modo no secuencial, desactive la opción *sequentialActions* en la sección MAIN.

En el modo secuencial, las acciones son procesadas en el orden en que fueron escritas, a excepción de ACTION-BOOT, la cual siempre se ejecuta al final, sin importar su ubicación en el archivo. Para usar este modo de ejecución, active la opción *sequentialActions* en la sección MAIN.

## Guía de acciones

Ésta es la lista completa de acciones:

ACTION-ZIP  
 ACTION-CPY  
 ACTION-ATTRIB  
 ACTION-REG  
 ACTION-DLL  
 ACTION-EXE  
 ACTION-WAIT-EXE  
 ACTION-REGEXP  
 ACTION-DEL  
 ACTION-REMOVE  
 ACTION-KILL  
 ACTION-DELAY  
 ACTION-BOOT

Exceptuando ACTION-BOOT y ACTION-DELAY (ver más adelante), las acciones presentan la misma estructura:

```
[action-name]
stopOnError = yes | no
filecount   = number of files (NN+1)
file00      = file and/or parameters
...
fileNN      = file and/or parameters
```

## Opción StopOnError:

Esta opción permite controlar la ejecución de las acciones: si esta opción se encuentra activada y la acción falla, las acciones subsiguientes no serán ejecutadas. Esto resulta útil cuando ciertas acciones dependen del éxito de las anteriores.

## ACTION-ZIP

### Descripción:

Descomprime archivos. Luego de descomprimido, el archivo **no** es borrado.

### Sintaxis:

fileXX = nombre del archivo

Si el camino o el nombre de archivo contienen espacios en blanco, use comillas. Los argumentos son opcionales y no requieren comillas.

### Ejemplo:

```
[action-zip]
filecount =2
file00 = \flash\application.zip
file01 = \configuration.zip
```

## ACTION-CPY

### Descripción:

Aplica un archivo cpy. Vea el apéndice "Creación de archivos cpy" para una descripción más detallada.

### Sintaxis:

fileXX = nombre del archivo

### Ejemplo:

```
[action-cpy]
filecount = 1
file00 = \flash\persist.copy
```

## ACTION-ATTRIB

### Descripción:

Cambia los atributos del archivo. Esta acción acepta wildcards ("?" y "\*").

### Sintaxis:

fileXX = nombre del archivo [-r | +r] [-h | +h]

Si el camino o el nombre de archivo contienen espacios en blanco, use comillas. Los argumentos son obligatorios y no requieren comillas.

### Parámetros:

+r/-r: activa/desactiva el atributo "sólo lectura"  
+h/-h: activa/desactiva el atributo "oculto"

### Ejemplo:

```
[action-attrib]
filecount = 4
file00 = \flash\myapp\file.txt -r
file01 = \flash\*.cfg +r
file02 = \myapp.exe -r -h
file03 = \flash\backup +r +h
```

## **ACTION-REG**

### **Descripción:**

Importa las claves del registro de un archivo.

### **Sintaxis:**

fileXX = nombre del archivo

No son necesarias las comillas para los caminos o nombres de archivos que contengan espacios en blanco.

### **Ejemplo:**

```
[action-reg]
filecount = 2
file00 = \flash\wifi.reg
file01 = \app.reg
```

## **ACTION-DLL**

No soportado

## **ACTION-EXE**

### **Descripción:**

Ejecuta un archivo ejecutable o un documento. Para poder ejecutar un documento, la extensión del archivo debe estar asociada a una aplicación.

Una vez realizada la ejecución, continúa con las acciones subsiguientes.

### **Sintaxis:**

fileXX = nombre del archivo

Si el camino o el nombre de archivo contienen espacios en blanco, use comillas. Los argumentos son opcionales y no requieren comillas.

**Ejemplo:**

```
[action-exe]
filecount = 1
file00 = \windows\app.exe

[action-exe]
filecount = 1
file00 = \windows\MyPage.htm

[action-exe]
filecount = 1
file00 = \windows\MyLink.lnk

[action-exe]
filecount = 1
file00 = "\Program Files\app.exe" -d MyArguments
```

**ACTION-WAIT-EXE****Descripción:**

Ejecuta un archivo ejecutable o un documento. Para poder ejecutar un documento, la extensión del archivo debe estar asociada a una aplicación.

Una vez realizada la ejecución, espera hasta que la aplicación se cierre antes de continuar con las acciones subsiguientes.

**Sintaxis:**

fileXX = nombre del archivo

Si el camino o el nombre de archivo contienen espacios en blanco, use comillas. Los argumentos son opcionales y no requieren comillas.

**Ejemplo:**

```
[action-wait-exe]
filecount = 1
file00 = \windows\app.exe
```

## ACTION-REGEXP

### Descripción:

Exporta la clave del registro a un archivo.

### Sintaxis:

fileXX = archivo [-a] reg-key

Si el camino o el archivo contienen espacios en blanco, use comillas.

Abreviaturas de las teclas principales:

HKLM : HKEY\_LOCAL\_MACHINE  
HKCU : HKEY\_CURRENT\_USER  
HKCR : HKEY\_CLASSES\_ROOT  
HKU : HKEY\_USERS

Para exportar la clave junto con sus subclaves use corchetes de esta manera:  
[HKLM\Comm]

Para exportar únicamente los valores de la clave use corchetes de esta otra manera:  
[HKLM]\Comm

No es posible exportar un solo valor de una clave.

### Parámetros:

-a: añadir el contenido de la clave del registro al final de archivo.

### Ejemplo:

```
[action-regexp]
filecount = 4
file00 = \myapp.reg [HKCU\software\myapp]
file01 = \devices.reg [HKLM\comm\device1]
file02 = \devices.reg -a [HKLM\comm\device2]
file03 = "\Program Files\devices.reg" -a [HKLM\my key\device2]
```

## ACTION-DEL

### Descripción:

Borra un archivo. Esta acción acepta wildcards ("?" y "\*").

### Sintaxis:

fileXX = nombre del archivo

No son necesarias las comillas para los caminos o nombres de archivos que contengan espacios en blanco.

### Ejemplo:

```
[action-del]
filecount = 3
file00 = \myapp.reg
file01 = \flash\1.txt
file02 = \doc\*.txt
```

## ACTION-REMOVE

### Descripción:

Borra carpetas y archivos. Esta acción acepta wildcards ("?" y "\*").

### Sintaxis:

fileXX = carpeta

### Ejemplo:

```
# remove a directory
[action-remove]
filecount = 1
file00 = \flash\test

# remove only the content of the directory
[action-remove]
filecount = 1
file00 = \flash\test\*.*
```

## ACTION-KILL

### Descripción:

Hace kill de un proceso en ejecución.

### Sintaxis:

fileXX = nombre del archivo ejecutable

El parámetro "file" puede ser o el nombre de un archivo ejecutable o el camino completo a un archivo ejecutable. En el caso de especificar un nombre de archivo, el cliente buscará entre los procesos en ejecución el primer proceso que tenga ese nombre de archivo.

Especifique un nombre de archivo sólo si está seguro de que es el único proceso con ese nombre ejecutándose en el sistema.

### Ejemplo:

```
[action-kill]
filecount = 1
file00 = \flash\myapp.exe
```

## ACTION-DELAY

### Descripción:

Realiza una pausa.

### Sintaxis:

```
[action-delay]
delay = duración de la pausa en milisegundos
```

### Ejemplo:

```
# delay 3 seconds
[action-delay]
delay = 3000
```

## ACTION-BOOT

### Descripción:

Realiza un warmboot o un coldboot. Esta acción puede aparecer una sola vez en un archivo y siempre es la última acción en ejecutarse.

### Sintaxis:

```
[action-boot]
type = warmboot | coldboot
immediate = yes | no
delay = delay-time in seconds
```

### Parámetros:

**immediate:** esta opción es válida sólo para archivos .wdp files (ver **WDP Files**). Si esta opción se encuentra en "yes", la acción ACTION-BOOT se ejecutará inmediatamente después de terminado el procesamiento del archivo wdp.

**delay:** esperar X segundos antes de hacer boot

### Ejemplos:

```
# warmboot
[action-boot]
type = warmboot
immediate = no

# coldboot
[action-boot]
type = coldboot
immediate = yes
```

## Apéndice A: creación de archivos CPY

Los archivos CPY contienen una lista de archivos a copiar.

### Sintaxis:

fuente > destino

fuente: camino completo al archivo fuente

destino: camino completo al archivo de destino

No se requieren comillas para el camino del archivo de fuente ni para el de destino.

### Ejemplo de archivo CPY:

```
\wdp\flexvnc.exe      >  \Windows\flexvnc.exe
\wdp\flexftpserver.exe >  \Windows\flexftpserver.exe
\wdp\flexrpvserver.exe >  \Windows\flexrpvserver.exe
```